EXPLORING THE STACKING STATE-SPACE

AGAPITO LEDEZMA, RICARDO ALER, and DANIEL BORRAJO
Universidad Carlos III de Madrid
Avda. de la Universidad, 30
28911 Leganés. Madrid (Spain)
ledezma@inf.uc3m.es, aler@inf.uc3m.es, dborrajo@ia.uc3m.es

Nowadays, there is no doubt that machine learning techniques can be successfully applied to data mining tasks. Currently, the combination of several classifiers is one of the most active fields within inductive machine learning. Examples of such techniques are boosting, bagging and stacking. From these three techniques, stacking is perhaps the less used one. One of the main reasons for this relates to the difficulty to define and parameterize its components: selecting which combination of base classifiers to use, and which classifier to use as the meta-classifier. One could use for that purpose simple search methods (e.g. hill climbing), or more complex ones (e.g. genetic algorithms). But before search is attempted, it is important to know the properties of the search space itself. In this paper we study exhaustively the space of Stacking systems that can be built by using four base learning systems: C4.5, IB1, Naive Bayes, and PART. We have also used the Multiple Linear Response (MLR) as meta-classifier. The properties of this state-space obtained in this paper will be useful for designing new Stacking-based algorithms and tools.

Keywords: Stacking, Meta-learning, Ensembles of classifiers.

1. Introduction

Nowadays, there is no doubt that machine learning techniques can be successfully applied to data mining tasks. A particularly successful approach is to combine classifiers to improve accuracy. The most important systems that have been proposed are bagging [1], boosting [2], and stacking [3]. Bagging uses majority vote to combine several classifiers obtained from different subsets of the data set. Boosting sequentially learns several classifiers, each focusing on the data that was misclassified by the previous classifier. All the classifiers are combined by weighted vote. Both bagging and boosting use the same learning algorithm to generate the ensemble of classifiers. Stacking learns how to combine the outputs of a set of classifiers that have been obtained by different learning algorithms. There are also many

variants that are becoming increasingly sophisticated, such as LPboosting [4] in the boosting subfield or Multiple Boosting [5] in the boosting-bagging subfield. There are also many variants of the basic stacking algorithm [6,7,8,9,10].

The main problem of stacking and any AI tool that needs to use it, is how to obtain the right combination of base classifiers and the meta-classifier. If the number of classifiers and algorithms to use is small, this problem can be solved by a simple method in a reasonable time: exhaustive search. For instance, if the goal is to build a stacking system made of three base classifiers and the meta-classifier, and there are four available learning algorithms, then only 16 stacking combinations need to be tested. If more classifiers are needed, then sampling techniques or heuristic search could be used instead of exhaustive search, in the same spirit as the wrapper approaches for attribute selection [11].

However, before search is used as the core of automatic configuration of stacking systems, it is important to know the properties of the state-space of stacking systems. In particular, it would be very useful to know the density of "good" stacking systems in these spaces and wether a particular meta-classifier allows to build more successful stacking configurations. This is estimated empirically in this paper.

We also want to empirically test the following hypothesis. In principle, the stacking meta-classifier can determine which base classifiers to take into account to reach the final decision (the base classifier outputs are the inputs to the meta-classifier), much in the same way as any learning algorithm can determine that some of its attributes are irrelevant (by not using them in the final hypothesis). If this is the case, using $n-1$ base classifiers should make no difference to using n classifiers, as the meta-classifier would learn that one of its n classifiers is irrelevant. We explore this issue in detail in the experimental evaluation section.

In this paper, we carry out an exhaustive study on the state-space of stacking systems with two, three, and four base classifiers that have been chosen from four well-known algorithms: C4.5 [12], IB1 [13], Naive Bayes [14], and PART [15]. Additionally, we also use the MLR (Multiple Linear Regression) like a meta-classifier [10]. We could have chosen many other very useful learning algorithms, such as neural networks. However, each experiment is very time consuming, and we have to bound the number of classifiers to be used. Since the results might be dependent on the set of chosen classifiers, we will explore in the future the effect of introducing other types of learning algorithms.

There are many ways to apply the general idea of stacked generalization. Merz [9] performs a correspondence analysis over a set of base models to choose uncorrelated models. LeBlanc and Tibshirani [8] analyze the stacked generalization with some regularization (non-negative constraint) to improve the prediction performance on one artificial dataset. Other works on stacked generalization have developed different focus [6,7,16]. Ting and Witten [10] use probability outputs from level-0 models instead a of class prediction as inputs to the level-1 model. They also study empirically which is the best meta-classifier in several domains but use only 3 base classifiers. In previous work [17], we extend Ting and Witten's work by exhaustively exploring

all the stacking configurations, using two, three, and four base classifiers.

This paper is organized as follows. Section 2 gives some background on stacking and explains how to explore the state-space of stacking systems. Section 3 describes the experimental setup and the experimental results, respectively. Finally, Section 4 discusses those results, and Section 5 draws some conclusions.

2. Stacking

Stacking is the abbreviation to refer to Stacked Generalization [3]. The main idea of stacking is to combine classifiers from different learners such as decision trees, instance-based, bayesian or rule-based learners. Since each one uses different knowledge representation and different learning biases, the hypothesis space will be explored differently, and different classifiers will be obtained. Thus, it is expected that their errors will not be correlated, and that the combination of classifiers will perform better than the base classifiers.

Once the classifiers have been generated, they must be combined. Stacking uses the concept of meta learner. The meta learner (or level-1 model) tries to learn how the decisions of the base classifiers (or level-0 models) should be combined to obtain the final classification. More formally, given a data set S, stacking first generates a subset of training sets $S_1, ..., S_T$ and then follows something similar to a cross-validation process: it leaves one of the subsets out (e.g. S_j) to use later. The remaining instances $S - S_j$ are used to generate the level-0 classifiers by applying K different learning algorithms, $k = 1, ..., K$, to obtain K classifiers. After the level-0 models have been generated, the S_j set is used to make the training set for the meta learner (level-1 classifier). Level-1 training data is built from the predictions of the level-0 models over the instances in S_j. Level-1 data has K attributes, whose values are the predictions of each one of the K level-0 classifiers for every instance in S_j, and the target class; i.e. the right class for every particular instance in S_j. Once the level-1 data has been built from all instances in S after the internal cross-validation process, any learning algorithm can be used to generate the level-1 model. To complete the process, the level-0 models are re-generated from the whole data set S (this way, it is expected that classifiers will be slightly more accurate). To classify a new instance, the level-0 models produce a vector of predictions that is the input to the level-1 model, which in turn predicts the class.

One of the main difficulties in applying this technique consists on identifying which learning techniques to use in the 0- and 1- levels. In this paper, the whole state-space of stacking systems with $i = 2$, 3, and 4 base classifiers will be studied. Base classifiers are chosen from a set that contains C4.5, IB1, PART, and Naive Bayes. The 1-level classifier is selected from the same set, plus the MLR meta-classifier. Once built, each resulting stacking system is tested with a testing set. In general, if b base classifiers can be chosen from n learning algorithms and there are m possible meta-classifiers, the number of stacking systems that can be built is $N = \binom{n}{b} * m$. In this paper, three sets of experiments have been carried out, with

$n = 4$, $m = 5$, and $b = 2$, $b = 3$, and $b = 4$, resulting in 30, 20, and 5 combinations, respectively. This is the space of stacking systems we are going to explore in this article.

3. Experiments and Results

From the many alternatives for inductive techniques, in this work we have used the algorithms implemented in the Waikato Environment for Knowledge Analysis, WEKA [18]. This software includes all the learning algorithms that we have used to build the base classifiers and an implementation of Stacked Generalization (stacking) that use probability outputs from level-0 models instead a simple class prediction as inputs to the level-1 model [10]. We selected four learning algorithms to build the stacking system:[1]

- C: C4.5 [12]. We used the version that generates decision trees.
- R: PART [15]. It generates a decision list from pruned partial decision trees generated using the C4.5 heuristic.
- N: A probabilistic Naive Bayesian classifier [14].
- I: IB1. Aha's instance based learning algorithm [13].
- M: MLR. This Multiple Linear Regression classifier was successfully used as a meta-classifier in [10].

For the experimental test of the stacking system configuration we have used eight data sets from the well known repository of machine learning databases at UCI [19]. These data sets have different sizes and include both nominal and numeric values. Table 1 shows the datasets features. In all the experiments we carry out ten-fold cross-validation. Thus, all the results shown in this paper are the average of the cross-validation process. In order to test whether differences are significant, every system has been compared to every other system with a paired t-test at a 0.05 confidence level.

The results obtained in the first set of experiments are shown in Table 2. In this set of experiments we used two base classifiers, thus obtaining 30 stacking systems by the combination of the four learning algorithms available. The best results in terms of accuracy are given in bold face.

In the second set of experiments we increased the number of base classifiers from two to three, resulting in 20 stacking systems. Table 3 shows the results obtained from this set of experiments. Also, in six of the eigth datasets the MLR meta-classifier perform better than any other meta-classifier. Those results are consistent with [10]. Previous work [17] shows that Naive Bayes is also a good meta-classifier. In

[1]For experimental purposes only default setting for all learning algorithms have been used.

Table 1: Descriptions of the used datasets.

Dataset	Attributes	Attributes Type	Instances	Classes
Heart	13	Numeric-nominal	303	2
Sonar	60	Numeric	208	2
Musk	166	Numeric	476	2
Ionosphere	34	Numeric	351	2
Horse-colic	23	Numeric-nominal	368	2
Breast-cancer	10	Nominal	286	2
Vowel	14	Numeric-nominal	990	11
hepatitis	20	Numeric-nominal	155	2

S3, differences between the best MLR and the best Naive Bayes configurations are not significant.[2]

In the other two domains, Naive Bayes is the best.

In the last set of experiments we used four base classifiers (5 stacking configurations). The results obtained from these experiments are shown in Table 4. Also, in six of the eigth datasets the MLR meta-classifier performs better than any other meta-classifier. Again, these results are consistent with [10]. In the other domain, Naive Bayes is the best meta-classifier. However, differences between MLR and Naive Bayes are not significant, except in the Vowel and Musk domains.

Table 5 summarizes previous results. In 4 of 8 domains, S2 finds the best stacking system. S3 wins S2 in 3 domains. In 5 of the 8 domains, S3 wins over S4. S4 wins over S3 in 1 domain. However, none of these differences are significant. Table 5 also provides results for the four algorithms used as standalone learning algorithms. C4.5-Bagging and C4.5-boosting results are also given for comparison purposes. The number of classifiers in bagging and boosting systems was set to 10 (boosting and bagging of C4.5 with this settings has shown good results in the literature [20]).

In order to summarize the whole stacking state-spaces, we have used cumulative probability graphs. They are shown in Figures 1, 2, 3, 4, 5, 6, 7 and 8 (one figure for each domain). These graphics give the probability (y-axis) of obtaining a stacking system with a testing accuracy equal or better than some value (in the x-axis). The accuracies for the best base classifier (BC), boosting, and bagging are displayed as vertical lines.

4. Discussion

Table 6 summarizes the best results obtained by each of the three main groups of classifiers used in this paper: base classifiers, stacking combinations, and bagging/boosting. Also, the difference between the best and worst results in the table

[2]It has to be taken into account that in this paper we use a paired t-test, whereas Ting & Witten used a 2σ test.

Table 2: Accuracies rates of stacking systems with two base classifiers (S2).

MC	BC	SONAR	HEART	MUSK	IONO	COLIC	VOWEL	CANCER	HEPATITIS
C	C-I	78.57	78.00	**86.67**	90.02	85.34	98.89	73.81	80.00
	C-N	76.19	83.33	82.08	90.89	83.69	81.31	71.01	80.08
	C-R	79.04	77.66	82.08	90.88	84.24	76.57	72.72	78.07
	I-N	78.10	83.00	86.25	90.03	81.27	98.48	72.08	80.04
	I-R	**80.00**	77.33	85.00	89.18	83.43	**98.99**	70.00	76.04
	N-R	76.19	82.33	81.46	89.75	82.06	79.09	71.70	79.42
I	C-I	59.52	49.67	75.83	86.31	62.12	96.97	71.03	76.83
	C-N	66.19	74.67	76.04	88.05	79.34	78.79	65.43	82.63
	C-R	62.86	65.00	70.21	85.47	65.69	75.96	68.20	71.62
	I-N	73.33	78.33	78.12	87.20	70.93	98.69	68.82	74.87
	I-R	59.52	61.33	76.88	84.89	70.00	96.26	64.00	72.87
	N-R	70.00	76.67	74.37	85.18	79.10	76.77	65.31	79.88
M	C-I	79.52	77.33	86.25	90.60	85.07	**98.99**	73.12	78.08
	C-N	78.10	83.67	82.08	90.60	85.33	79.09	69.59	84.58
	C-R	79.52	78.67	81.67	**91.74**	83.72	79.60	72.08	78.75
	I-N	79.52	**85.33**	86.25	86.89	82.09	**98.99**	71.66	80.71
	I-R	79.05	78.00	84.79	90.61	83.99	**98.99**	72.09	79.96
	N-R	76.67	83.67	81.67	91.18	83.99	78.89	70.28	**85.17**
N	C-I	**80.00**	80.33	82.50	90.31	**85.61**	80.10	**75.18**	78.79
	C-N	78.10	82.67	82.29	90.88	83.44	78.48	72.36	81.25
	C-R	79.05	79.67	81.46	91.18	84.80	74.85	73.78	82.63
	I-N	70.48	84.33	81.04	83.77	78.84	77.07	72.33	84.42
	I-R	76.67	78.33	81.67	91.18	83.45	80.00	72.06	80.67
	N-R	75.24	83.67	81.67	91.18	83.18	76.67	70.60	82.46
R	C-I	78.57	78.00	86.46	90.31	85.34	98.89	73.81	79.38
	C-N	77.62	83.33	82.08	90.32	84.23	81.01	71.01	80.71
	C-R	79.05	78.00	81.46	90.88	83.96	78.28	71.39	78.04
	I-N	78.10	83.33	86.25	89.75	81.01	98.28	72.08	79.42
	I-R	**80.00**	77.00	85.00	89.18	83.16	**98.99**	70.00	75.42
	N-R	76.67	82.33	81.46	90.03	82.06	80.51	71.70	79.42

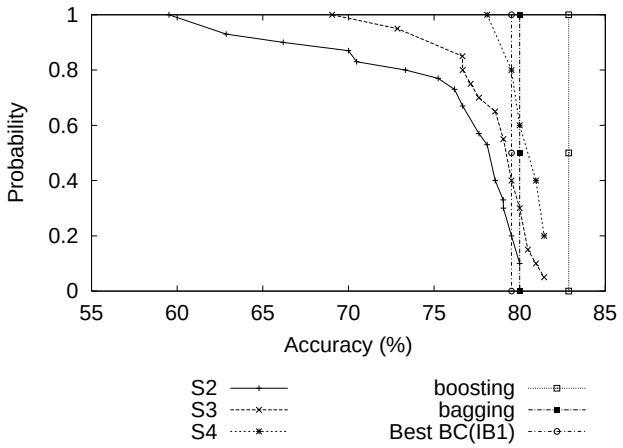

Figure 1: Cumulative probability in the SONAR domain.

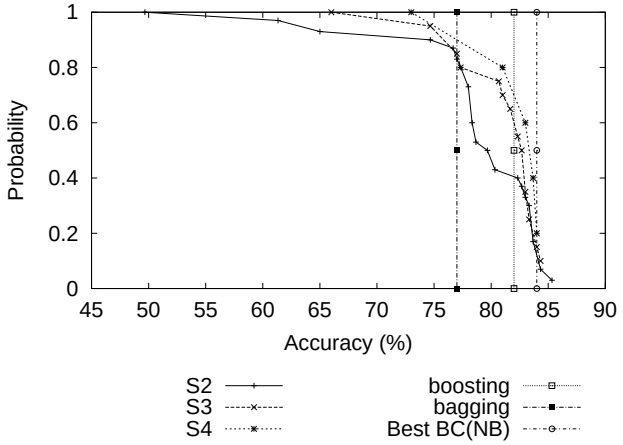

Figure 2: Cumulative probability in the HEART domain.

Table 3: Accuracy rates of stacking systems with three base classifiers (S3).

MC	BC	SONAR	HEART	MUSK	IONO	COLIC	VOWEL	CANCER	HEPATITIS
C	C-I-N	78.57	83.00	86.88	90.89	83.69	98.59	71.06	79.38
	C-I-R	79.05	77.00	85.83	91.74	83.43	98.89	72.41	77.96
	C-N-R	77.14	82.67	82.08	92.03	82.32	81.21	73.10	78.75
	I-N-R	78.57	83.33	86.46	89.74	81.52	98.79	71.38	78.04
I	C-I-N	72.86	74.67	83.13	88.33	77.70	96.36	64.33	77.46
	C-I-R	69.05	66.00	79.37	89.17	70.29	93.43	66.07	70.96
	C-N-R	72.86	74.67	77.50	88.62	78.55	78.99	62.20	77.29
	I-N-R	76.67	77.33	82.71	86.61	77.45	96.36	65.32	73.42
M	C-I-N	80.00	**84.33**	**87.50**	91.74	**85.33**	**98.99**	70.28	82.00
	C-I-R	80.00	80.67	87.08	92.88	84.26	**98.99**	72.77	79.96
	C-N-R	79.52	83.00	82.29	93.17	84.80	82.02	69.59	**84.58**
	I-N-R	80.00	84.00	88.13	**93.45**	83.99	98.99	71.31	81.96
N	C-I-N	80.48	82.67	84.58	91.17	83.45	82.73	72.02	82.54
	C-I-R	**81.43**	81.00	85.21	91.18	84.80	80.51	**74.47**	83.88
	C-N-R	79.52	82.33	82.71	92.04	84.52	81.82	71.70	82.54
	I-N-R	80.95	83.33	84.58	92.33	82.36	81.72	70.58	83.13
R	C-I-N	79.05	81.67	86.67	90.60	83.97	98.59	71.05	78.13
	C-I-R	79.52	77.00	83.65	89.45	83.43	98.59	73.46	76.71
	C-N-R	77.62	82.67	82.08	92.32	83.70	82.42	71.01	79.38
	I-N-R	79.05	**84.33**	86.46	91.17	81.79	98.69	71.38	76.75

is shown in the fourth column. As previous research suggests, the best results are always obtained by the ensemble of classifiers systems (either stacking or boosting) against the single inductive learning techniques. The largest difference between the best and the worse configuration is 6.33. This is because the best base classifier in this domain is IB1. The stacking systems that use it in the base are frequently very good. Boosting uses C4.5, which is not very good in this domain, hence the large difference between stacking and boosting. Otherwise, such differences are not very large (3% on average). This is important to remark because in many cases, significance is highlighted, even if the actual difference is small.

Our first issue is to determine the density of "good" stacking configurations in the stacking state space. That is, if we were to draw randomly a stacking system, what would be the probability of it being a good one. Table 7 displays the percentage of stacking systems which are significantly better or worse than boosting, or not significantly different from it (column "equal"). It can be seen that only in one domain, there is a significantly large number of stacking systems better than

Table 4: Accuracy rates of stacking systems with four base classifiers (S4).

MC	BC	SONAR	HEART	MUSK	IONO	COLIC	VOWEL	CANCER	HEPATITIS
C	C-I-N-R	79.52	83.00	86.67	90.60	82.33	98.69	71.39	80.00
I	C-I-N-R	78.10	73.00	84.58	88.62	77.18	94.14	62.89	72.12
M	C-I-N-R	**81.43**	**84.00**	**88.12**	**93.17**	84.53	**98.99**	69.93	**83.25**
N	C-I-N-R	80.95	83.67	83.96	92.33	**85.07**	83.94	**71.67**	81.21
R	C-I-N-R	80.00	81.00	86.67	90.03	82.33	98.38	69.31	76.75

Table 5: Accuracy rates of base classifiers and the best stacking systems.

DOMAIN	C4.5	IB1	NBAYES	PART
SONAR	78.57	79.52	67.14	76.67
HEART	80.33	78.67	84.00	77.67
MUSK	81.88	86.25	73.54	81.67
IONO	90.31	86.89	83.20	91.18
COLIC	85.88	79.08	79.10	82.64
VOWEL	78.38	98.99	61.52	77.37
CANCER	75.18	65.41	74.06	71.37
HEPATITIS	79.42	79.96	83.79	80.63

	BEST S2	BEST S3	BEST S4	BAGGING/BOOSTING (WITH C4.5)
SONAR	80.00	81.43	81.43	80.00/82.86
HEART	85.33	84.33	84.00	77.00/82.00
MUSK	86.67	87.50	88.12	88.75/89.38
IONO	91.74	93.45	93.17	90.88/93.73
COLIC	85.61	85.33	85.07	85.61/82.09
VOWEL	98.99	98.99	98.99	88.59/92.22
CANCER	75.18	74.47	71.67	73.07/67.48
HEPATITIS	85.17	84.58	83.25	86.46/82.63

Boosting (although in that case, there is also a large number of stacking systems worse than Boosting). See column $S2 \cup S3 \cup S4$ in Table 7 for a summary. In the rest of domains, either most of the stacking systems are comparable to Boosting, or worse than it. Therefore, it seems that in most cases, there is a high density of stacking systems comparable to Boosting, and in some cases, there is a large probability that the configuration will be worse than Boosting.

As we have said previously, the best stacking systems are usually obtained by MLR, although Naive Bayes is also a good candidate. However, what is the percentage of stacking systems which are good if MLR is used?. To answer this question, Table 8 displays the percentage of stacking systems that are significantly better, worse or not significantly different than Boosting. Results are broken down according to the meta-classifier. In four domains, using MLR as the meta-classifier

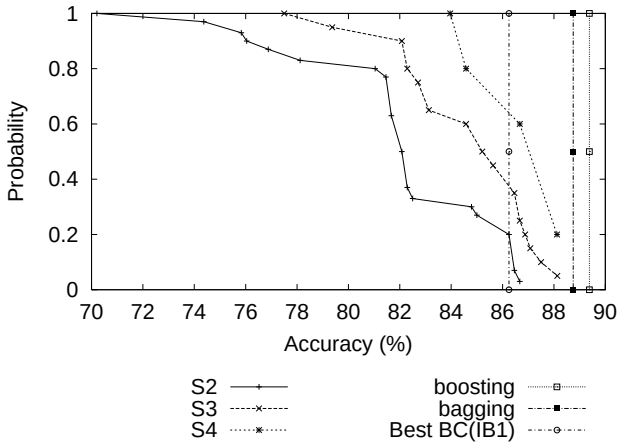

Figure 3: Cumulative probability in the MUSK domain.

Table 6: Best results from the three following groups: base classifiers, stacking combinations, and boosting/bagging with C4.5.

DOMAIN	BEST BASE CLASSIFIER	BEST STACKING	BAGGING/ BOOSTING(C4.5)	DIFFERENCE BEST-WORSE
SONAR	IB1 (79.52)	S3/S4 (81.43)	(**82.86**)	3.34
HEART	NBAYES (84.00)	S2 (**85.33**)	(82.00)	3.33
MUSK	IB1 (86.25)	S4 (88.12)	(**89.38**)	3.13
IONOSPHERE	PART (91.18)	S3 (93.45)	(**93.73**)	2.55
COLIC	C4.5 (**85.88**)	S2 (85.61)	(85.61)	0.27
VOWEL	IB1 (**98.99**)	S2/S3/S4 (**98.99**)	(92.66)	6.33
CANCER	C4.5 (**75.18**)	S2 (**75.18**)	(73.07)	2.11
HEPATITIS	NBAYES (83.79)	S2 (85.17)	(**86.46**)	2.67

obtains results comparable to Boosting independently of the base that is used (no significant differences). But this is also true of C4.5. Naive Bayes, and PART get three domains. Only IB1 gets bad results in this respect. In the rest of domains, at least half of stacking configurations that use MLR are better than Boosting. This is also true for Naive Bayes, but not for the rest. So, it seems that using MLR or Naive Bayes as meta-classifier is good guarantee that the resulting configuration will be a good one.

Our next issue is whether a stacking configuration is able to improve over its best base classifier. It would be interesting that this happens often in order for stacking to be useful. Table 9 displays those results for S2 $\cup$ S3 $\cup$ S4. It can be seen that in most cases (except in the vowel domain), most stacking systems are not significantly different. But in some cases (musk and ionosphere domains), there is a large probability that the resulting stacking configuration will be significantly

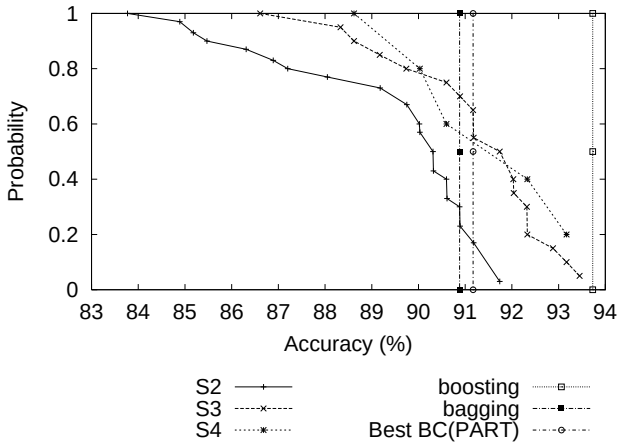

Figure 4: Cumulative probability in the IONOSPHERE domain.

Table 7: Percentage of stacking systems that are significantly better, worse or not significantly different than Boosting.

	S2			S3		
	Better	Equal	Worse	Better	Equal	Worse
colic	0.0%	89.66%	10.34%	0.0%	84.21%	15.79%
vowel	41.38%	0.0%	58.62%	52.63%	5.26%	42.11%
cancer	3.45%	96.55%	0.0%	0.0%	100.0%	0.0%
hepatitis	0.0%	89.66%	10.34%	0.0%	94.74%	5.26%
heart	0.0%	82.76%	17.24%	0.0%	94.74%	5.26%
musk	0.0%	24.14%	75.86%	0.0%	47.37%	52.63%
iono	0.0%	34.48%	65.52%	0.0%	47.37%	52.63%
sonar	0.0%	68.97%	31.03%	0.0%	78.95%	21.05%
	S4			S2 ∪ S3 ∪ S4		
colic	0.0%	80.0%	20.0%	0.0%	86.79%	13.21%
vowel	60.0%	20.0%	20.0%	47.17%	3.77%	49.06%
cancer	0.0%	100.0%	0.0%	1.89%	98.11%	0.0%
hepatitis	0.0%	80.0%	20.0%	0.0%	90.57%	9.43%
heart	0.0%	80.0%	20.0%	0.0%	86.79%	13.21%
musk	0.0%	60.0%	40.0%	0.0%	35.85%	64.15%
iono	0.0%	40.0%	60.0%	0.0%	39.62%	60.38%
sonar	0.0%	100.0%	0.0%	0.0%	75.47%	24.53%

worse than its best base classifier.

Finally, we want to determine whether adding extra classifiers in the base is usually a good thing. Table 10 displays the percentage of stacking configurations of S(i+1) that include the best stacking configuration of S(i) (i.e., same meta-classifier and the set of base classifiers of S(i) is a subset of the set of base classifiers of S(i+1)). This table shows that adding a new base classifier to the best stacking configuration does not improve accuracy significantly, but it does not perform worse either. Additional results show that the best S3 stacking configuration is not significantly better (nor worse) than the one in S2. Likewise for S4 and S3. Therefore, adding

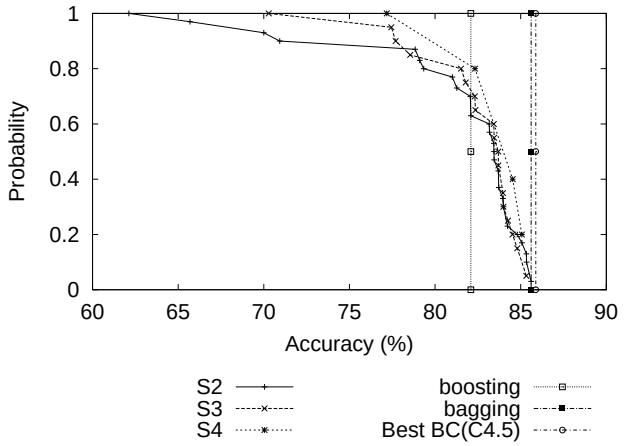

Figure 5: Cumulative probability in the HORSE-COLIC domain.

new base classifiers is not always good, but it does not worsen things either.

5. Conclusions

The aim of this paper was to systematically study the state-space of heterogeneous stacking systems. Here, we have studied empirically the state-space of stacking systems with 2, 3, and 4 base classifiers, that can be built using C4.5, PART, Naive Bayes, and IB1. Also, MLR has been used as a meta-classifier. As this state-space is not too large, it can be studied exhaustively. The most important conclusions of this paper are:

- The stacking state-space contains systems which are comparable to Boosting. This is important, because even though the computational effort of searching for the best stacking configuration is larger than for boosting, the state-space defined in this paper is small enough to be explored in a reasonable time. Also, only a few base classifiers are needed to get comparable results to boosting.

- However, the density of good stacking systems is not always high. However, if MLR or Naive Bayes are used, in the domains we have explored, at least 50% of the configurations that use them as meta-classifiers will be comparable or better than Boosting.

- With respect to the issue of whether a stacking configuration is able to improve upon its best base classifier, the conclusion is that in most cases, most stacking systems are not significantly different. But in some cases, there is a large probability that the resulting stacking configuration will be significantly worse than its best base classifier.

- Therefore, if larger state-spaces are to be searched (because we want to use

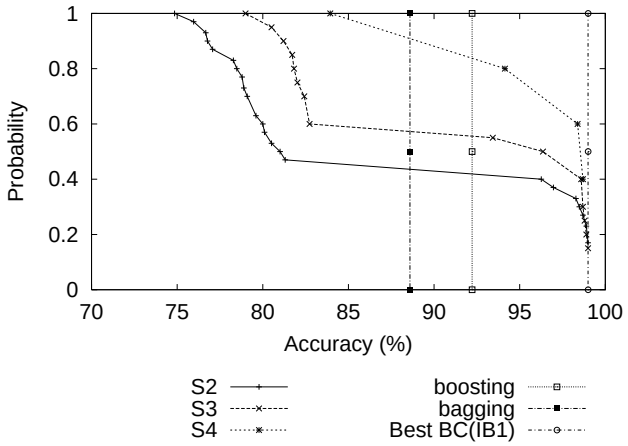

Figure 6: Cumulative probability in the VOWEL domain.

more base classifiers, for instance), heuristics will be needed to do so efficiently. For instance, our systematic study suggests that MLR seems to be the most appropriate meta-classifier. Also, simple heuristic methods like hill-climbing, simulated annealing, or genetic algorithms could be used. We have used genetic algorithms with good results in [21].

- We have also found out that merely increasing the number of base classifiers does not always pay off in terms of accuracy.

Bibliography

[1] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[2] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In Springer-Verlag, editor, *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, 1995.

[3] D. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[4] K. Bennett, A. Demiriz, and J. Shawe-Taylor. A column generation algorithm for boosting. In P. Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 65–72. Morgan Kaufmann, 2000.

[5] Z. Zheng and G.I. Webb. Multiple boosting: A combination of boosting and bagging. In *Proceedings of the 4th International Conference on Parallel and Distributed Processing Techniques and Applications*, pages 1133–1140. CSREA Press, 1998.

[6] P. Chan and S. Stolfo. A comparative evaluation of voting and meta-learning on partitioned data. In Morgan Kaufmann, editor, *Proceedings of Twelfth International Conference on Machine Learning*, pages 90–98, 1995.

[7] D. Fan, P. Chan, and S. Stolfo. A comparative evaluation of combiner and stacked generalization. In *Proccedings of AAAI-96 Workshop on Integrating Multiple Learning Models*, pages 40–46, 1996.

[8] M. LeBlanc and R. Tibshirani. Combining estimates in regression and classification. In *Technical Report 9318*. Deparment of Statistic, Univesity of Toronto, 1993.

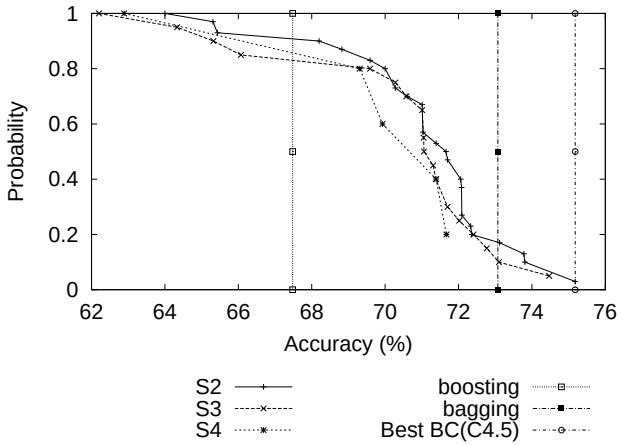

Figure 7: Cumulative probability in the BREAST CANCER domain.

[9] Christopher J. Merz. Using correspondence analysis to combine classifiers. *Machine Learning*, 36:33, 1999.

[10] K. Ting and I. Witten. Stacked generalization: when does it work? In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1997.

[11] R. Kohavi and G.H. John. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, 1995.

[12] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[13] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, jan 1991.

[14] G. John and P. Langley. Estimating continuous distribution in bayesian classifiers. In Morgan Kaufmann, editor, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

[15] E. Frank and I. Witten. Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 144–151. Morgan Kaufmann, 1998.

[16] D. Fan, S. Stolfo, and P. Chan. Using conflicts among multiple base classifiers to measure the performance of stacking. In *Proceedings of the ICML-99 Workshop on Recent Advances in Meta-Learning and Future Work*, pages 10–17, 1999.

[17] Ricardo Aler Agapito Ledezma and Daniel Borrajo. Empirical study of stacking state-space. In *Thirteenth International Conference on Tools with Artificial Intelligence, ICTAI'01*, pages 210–217, 2001.

[18] I. Witten and E. Frank. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 2000.

[19] C. Blake and C. Merz. Uci repository of machine learning databases. databases http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.

[20] J.R. Quinlan. Bagging, boosting, and c4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 725–730. AAAI Press / MIT Press, 1996.

[21] Agapito Ledezma, Ricardo Aler, and Daniel Borrajo. *Heuristic and Optimization for*

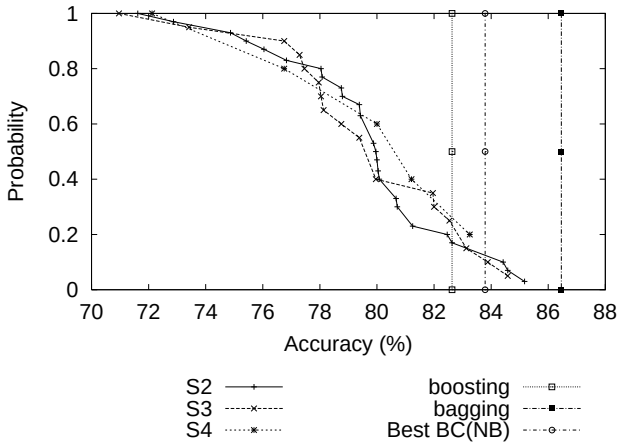

Figure 8: Cumulative probability in the HEPATITIS domain.

Knowledge Discovery, chapter Heuristic Search Based Stacking of Classifiers. Idea Group Publishing, 2001.

Table 8: Percentage of stacking systems that are significantly better, worse or not significantly different than Boosting. Results are broken down according to the meta-classifier.

		S2 ∪ S3 ∪ S4		
Domain	Meta	Better	Equal	Worse
colic	M	0.0%	100.0%	0.0%
colic	C	0.0%	100.0%	0.0%
colic	R	0.0%	100.0%	0.0%
colic	I	0.0%	36.36%	63.64%
colic	N	0.0%	100.0%	0.0%
vowel	M	63.64%	0.0%	36.36%
vowel	C	70.0%	0.0%	30.0%
vowel	R	60.0%	0.0%	40.0%
vowel	I	45.45%	18.18%	36.36%
vowel	N	0.0%	0.0%	100.0%
cancer	M	0.0%	100.0%	0.0%
cancer	C	0.0%	100.0%	0.0%
cancer	R	0.0%	100.0%	0.0%
cancer	I	0.0%	100.0%	0.0%
cancer	N	9.09%	90.91%	0.0%
hepatitis	M	0.0%	100.0%	0.0%
hepatitis	C	0.0%	100.0%	0.0%
hepatitis	R	0.0%	100.0%	0.0%
hepatitis	I	0.0%	54.55%	45.45%
hepatitis	N	0.0%	100.0%	0.0%
heart	M	0.0%	90.91%	9.09%
heart	C	0.0%	100.0%	0.0%
heart	R	0.0%	90.0%	10.0%
heart	I	0.0%	54.55%	45.45%
heart	N	0.0%	100.0%	0.0%
musk	M	0.0%	54.55%	45.45%
musk	C	0.0%	60.0%	40.0%
musk	R	0.0%	50.0%	50.0%
musk	I	0.0%	18.18%	81.82%
musk	N	0.0%	0.0%	100.0%
iono	M	0.0%	63.64%	36.36%
iono	C	0.0%	20.0%	80.0%
iono	R	0.0%	30.0%	70.0%
iono	I	0.0%	18.18%	81.82%
iono	N	0.0%	63.64%	36.36%
sonar	M	0.0%	100.0%	0.0%
sonar	C	0.0%	90.0%	10.0%
sonar	R	0.0%	100.0%	0.0%
sonar	I	0.0%	9.09%	90.91%
sonar	N	0.0%	81.82%	18.18%

Table 9: Percentage of stacking systems that are significantly better, worse or not significantly different than the best classifier in its base.

	S2 ∪ S3 ∪ S4		
Domain	Better	Equal	Worse
colic	0.0%	86.54%	13.46%
vowel	47.17%	3.77%	49.06%
cancer	1.92%	98.08%	0.0%
hepatitis	0.0%	90.57%	9.43%
heart	0.0%	86.79%	13.21%
musk	0.0%	35.85%	64.15%
iono	0.0%	39.62%	60.38%
sonar	0.0%	75.47%	24.53%

Table 10: Percentage of stacking systems of S(i+1) that are significantly better, worse or not significantly different than the best stacking configuration of S(i), when S(i) is in S(i+1).

	S3 vs. best of S2			S4 vs. best of S3		
	Better	Equal	Worse	Better	Equal	Worse
colic	0.0%	100.0%	0.0%	0.0%	100.0%	0.0%
vowel	0.0%	100.0%	0.0%	0.0%	100.0%	0.0%
cancer	0.0%	100.0%	0.0%	0.0%	100.0%	0.0%
hepatitis	0.0%	100.0%	0.0%	0.0%	100.0%	0.0%
heart	0.0%	100.0%	0.0%	0.0%	100.0%	0.0%
musk	0.0%	100.0%	0.0%	0.0%	100.0%	0.0%
iono	0.0%	100.0%	0.0%	0.0%	100.0%	0.0%
sonar	0.0%	100.0%	0.0%	0.0%	100.0%	0.0%